International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 5 Issue 4, May-June 2021 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

LSTM Based Sentiment Analysis

Dirash A R[1], Dr. S K Manju Bargavi[2]

[1]Student, [2]Professor,
[1,2]Department of MCA, School of CS & IT, Jain Deemed-to-be University, Bangalore, Karnataka, India

ABSTRACT

Sentimental analysis is a context-based mining of text, which extracts and identify subjective information from a text or sentence provided. Here the main concept is extracting the sentiment of the text using machine-learning techniques such as LSTM (Long short-term memory). This text classification method analyses the incoming text and determines whether the underlined emotion is positive or negative along with probability associated with that positive or negative statements. Probability depicts the strength of a positive or negative statement, if the probability is close to zero, it implies that the sentiment is strongly negative and if probability is close to1, it means that the statement is strongly positive. Here a web application is created to deploy this model using a Python-based micro framework called flask. Many other methods, such as RNN and CNN, are inefficient when compared to LSTM.

***KEYWORDS:** Flask, IDMB, LSTM, Machine Learning, RNN*

***How to cite this paper**:* Dirash A R | Dr. S K Manju Bargavi "LSTM Based Sentiment Analysis" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-5 | Issue-4, June 2021, pp.728-732, URL: www.ijtsrd.com/papers/ijtsrd42345.pdf

IJTSRD42345

I. INTRODUCTION

Sentiment Analysis is a type of predictive modelling activity that involves training a model to predict the polarity of textual data or sentiments such as positive and negative. Various companies use Sentimental Analysis to better understand their customers' reactions to their goods.The ability of algorithms to analyze text has greatly improved as a result of recent developments in deep learning.

In the early stages, methods such as Naive Bayes, Support Vector Machines, and others are used to classify sentiment. Neural networks and other deep learning approaches (CNN, RNN, ANN) have recently acquired prominence due to their outstanding results. The dataset is taken from the IMDB movie review dataset. Movie reviews also provide valuable information about the film. By reading these reviews, we can manually determine if a film is good or bad. These Positive or negative sentiment analysis supports companies in determining the social sentiments of their brands, services, or goods based on online conversations on Twitter, Facebook, and other social media platforms.

Users may provide text reviews, comments, or suggestions to products on several social networking platforms or e-commerce websites. These user-generated texts are an good source of user sentiment on a variety of products and services. For a product, such text has the potential to expose both the item's relevant function/aspects as well as the users' feelings about each feature. The item's feature/aspects listed in the text serve the same purpose as meta-data in content-based filtering, but the former are more relevant to the recommender framework.. Since these features are often listed by users in their reviews, they can be considered the most important features that can have a major impact on the user's experience with the item, while the item's meta-data (which is typically provided by manufacturers rather than consumers) may overlook features that are important to users. A consumer can express different emotions for different things with similar characteristics. Also, different users can have different feelings about the same function of an object. Users reactions to features can be represented by a multi-dimensional rating score that reflects their preferences.

The work presented in this paper focuses on sentiment analysis using a machine learning approach. We can classify reviews based on emotion in a variety of ways, but we are using the most modern technique, LSTM networks. The model can predict sentiment analysis on text using this approach, and it is very accurate.

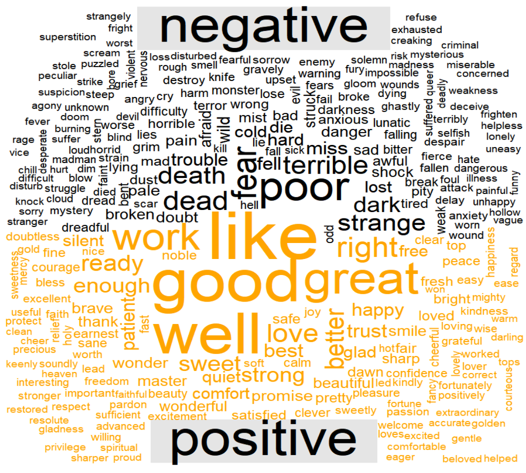

Fig: 1 List of positive and negative words

II. RELATED WORKS

Deep learning algorithms have shown to be effective in a variety of applications, including speech recognition, pattern recognition, and data classification. These approaches use a deeper hierarchy of structures in neural networks to learn data representation. Complicated concepts can be learned using simpler ones as a foundation. Convolutional Neural Networks (CNNs) have been shown to learn local features from words or phrases among deep feed forward networks [1].

Although Recurrent Neural Networks (RNNs) can learn temporal dependencies in sequential data, they can't learn temporal dependencies in random data[2].There aren't many features in social media because the messages are so short. To obtain more useful features, we expand on the concept of distributed representation of terms, in which each input is expressed by a large number of features, each of which is involved in a large number of possible inputs. We employ the Word2Vec word embedding model [3] to represent social posts in a distributed manner.

[4]In this paper, we wish to examine how effective long short-term memory (LSTM) [4] is in categorising sentiment on brief texts in social media with scattered representation. To begin, words in short texts are represented as vectors using a Word2Vec-based word embedding model. Second, in short texts, LSTM is used to learn long-distance dependence between word sequences. The prediction outcome is based on the final output from the previous point in time. In our sentiment classification studies on different social datasets, we compared the efficiency of LSTM with Nave Bayes (NB) and Extreme Learning Machine (ELM). As the results of the experiments demonstrate, our proposed approach outperforms traditional probabilistic models and neural networks with more training data.

An artificial neural network is a network structure inspired by human brain neurons. Nodes are arranged into layers, with edges connecting nodes in adjacent layers. Errors can be sent back to previous layers to adjust the weights of corresponding edges via feed-forward computations. Extreme Learning Machines (ELMs[5]are a form of neural network that does not use back propagation to change the weights. The secret nodes are allocated at random and are never updated. As a result, the weights are normally learned in a single move, which saves time.

Deep learning techniques, which employ several hidden layers, are used for more complicated relationships. It typically takes longer to compute with deeper network structures. These methods were made possible by recent advancements in hardware computing power and GPU processing in software technologies. Several forms of deep neural networks have been proposed based on various ways of structuring multiple layers, with CNNs and RNNs being the most common. Convolution operations are naturally applied in edge detection and image sharpening, so CNNs are commonly used in computer vision. They can also be used to compute weighted moving averages and impulse responses from signals. RNNs are a form of neural network in which the current inputs of hidden layers are determined by the previous outputs of hidden layers. This allows them to interact with time sequences containing temporal connections, such as speech recognition. In a previous comparative research of RNN vs CNN in natural language processing, RNNs were found to be more successful in sentiment analysis than CNNs [6]. As a result, the focus of this study is on RNNs.

Weights in RNNs may grow out of reach or disappear as the time series grows longer. To overcome the vanishing gradient problem [7]in training typical RNNs, LSTM [4]was proposed to learn long-term dependence across extended time periods. LSTM includes forget gates in addition to input and output gates. They're often used in applications including time series prediction and handwriting recognition. In this research, we use LSTM to build sentiment classifiers for shorter texts.

Natural language processing benefits from examining the distributional relationships between word occurrences in documents. The most straightforward method is to use one-hot encoding to describe each word's occurrence in documents as a binary vector. Word embedding models are used in distributional semantics to map from a one-hot vector space to a continuous vector space with a smaller dimension than the traditional bag-of-words model. The most common word embedding models are distributed representations of words, such as Word2Vec [2] and GloVe[8]which use neural networks to train occurrence relations between words and documents in the contexts of training data. In this research, we employ the Word2Vec word embedding model to describe words in short texts. Then, to catch the long-term dependence among words in short texts, LSTM classifiers are educated. Each text's sentiment can then be graded as either positive or negative.

Aditya Timmaraju and Vikesh Kumar [9] suggested a Recursive RNN-based intelligent model for classifying movie reviews. This is a method for emotion detection in natural language, such as text. This framework is an add-on for sentiment classification at the sentence stage.

JanDeriu and Mark Ceilieba[10] have developed a framework that categorizes the sentiment of tweets. The deep learning methodology is used in this research. They used 2-layer convolution neural networks in this study. The entire role is divided into three subtasks here.

Sachin Kumar and Anand Kumar [11] proposed a new method for the same mission, but using a different methodology, namely CNN. Sentiment analysis in NLP is used in this paper since most texts contain knowledge in the form of thoughts and emotions. This model will include a detailed analysis of an opinion or sentiment that can be classified as negative, optimistic, or neutral.

III. WORKING METHEODOLOGY

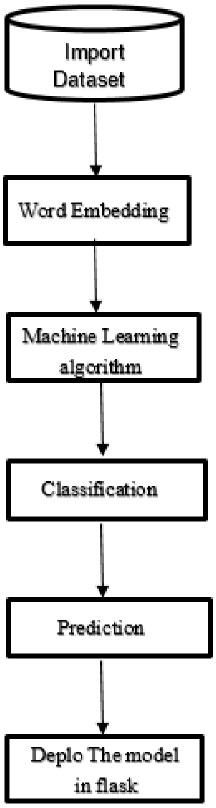

Fig: 2 Architectural diagram of classification process

3.1. Import Dataset

The model is based on the IMDB dataset, which stands for "Internet Movie Database", it is owned by Amazon. It has information related to Movies, Video games, Web Series, TV Series etc. that can be downloaded using the keras. dataset in a format that is ready to use for neural networks. This data contains 25000 movie reviews from IMDB, all of which have already been reprocessed and labelled as either positive or negative.

3.2. Word Embedding

In short texts, there is a better representation of minimal content; we used the embedding layer, which is one of the keras module's layers used for this purpose. Word embedding is a text mining approach that develops a link between words in textual data (Corpus).The context in which words are used determines their syntactic and semantic meanings. The distributional hypothesis posits that words with similar semantic meanings emerge in similar settings.

3.3. Machine Learning Algorithm

The dataset is divided into training dataset and Test set.We will build a neural network model to solve a basic sentiment analysis problem. The LSTM algorithm is used to build a model for classifying sentiment analysis. LSTM stands for Long short term memory. They are type of RNN (**Recurrent neural network**) which is well suited for sequence prediction problem. We can classify feedback based on emotion in a number of ways, but we're using LSTM networks, which is the most recent technique. Using this method, the model can predict sentiment analysis on text, and it is very accurate.

3.4. Classification

Google Collaboration will be used to train the model. The model is trained using the training dataset before being put to the test in the following phases. The accuracy of the qualified sentiment classification model is determined using the test dataset. The accuracy of the model determines the model's efficiency.

3.5. Prediction

This text classification method examines the input text and decides whether the underlined emotion is positive or negative, as well as the likelihood associated with certain positive or negative statements. The intensity of a positive or negative argument is represented by probability. The model's accuracy is 86.68 percent while using the LSTM networks algorithm.

3.6. Deploy the Model In Flask

Following the development of the model, the next step is to create a Web Application for it, which will be done with Flask. Flask is a Python based micro web open source Framework which doesn't require a particular tools or libraries. Flask gives you the tools, frameworks, and technologies needed to create a web application.

IV. RESULT

In this model, learning is the first step, and predicting is the second. The model is trained with the dataset in the learning process, and it classifies the train dataset according to that perception. During the learning process, the model is trained with the dataset, and it then classifies the train dataset according to that perception. To avoid under fitting, we should use a large dataset and a well-completed learning process. Based on the training, the model learns how to classify, and the model is then evaluated with the test set during the testing process.

```
________________________________________________
None
Epoch 1/30
391/391 [==============================] - 17s 39ms/step - loss: 0.5866 - accuracy: 0.6543 - val_loss: 0.3465 -
Epoch 2/30
391/391 [==============================] - 15s 38ms/step - loss: 0.2874 - accuracy: 0.8856 - val_loss: 0.3222 -
Epoch 3/30
391/391 [==============================] - 15s 38ms/step - loss: 0.2555 - accuracy: 0.9017 - val_loss: 0.3300 -
Epoch 4/30
391/391 [==============================] - 15s 38ms/step - loss: 0.3846 - accuracy: 0.8231 - val_loss: 0.3536 -
Epoch 5/30
391/391 [==============================] - 15s 38ms/step - loss: 0.2406 - accuracy: 0.9057 - val_loss: 0.3121 -
Epoch 6/30
```

Fig 3 Epoch is generated

Fig 3 shows the results of the learning phase. Here tenser flow is used, which is the backend of this model, which helps in various machine learning task. The model is trained using the training dataset before being put to the test in the following phases. We can see how the output improves over time as the epochs pass. The model's efficiency improves epoch by epoch, meaning that it learns from its experience.

```
#Calculate Accuracy
scores = model.evaluate(X_test, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

Accuracy: 85.68%
```

Fig 4 Accuracy is generated

Using the LSTM networks algorithm, the model's accuracy is 86.68%, as shown in Fig 2. When we use this dataset, the performance of this model outperforms all other machine learning algorithms. The LSTM is exceptional at classifying sequence text data.

V. OUTPUT

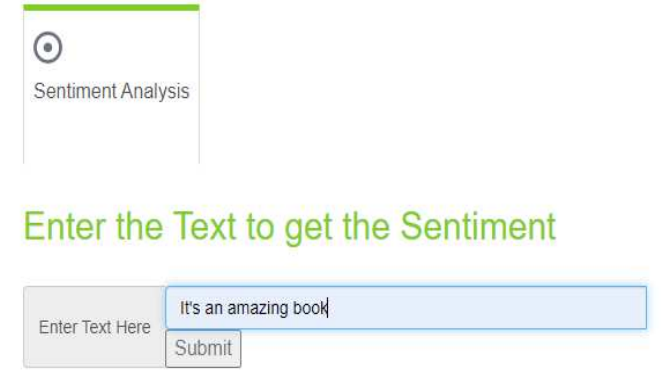

Fig 5 Text is entered to check Sentiment

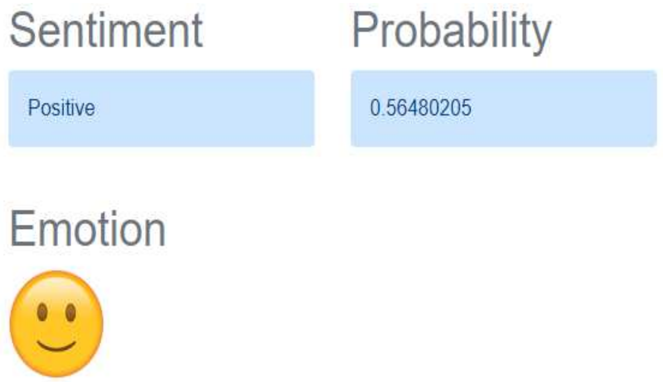

Fig 6 Sentiment is generated

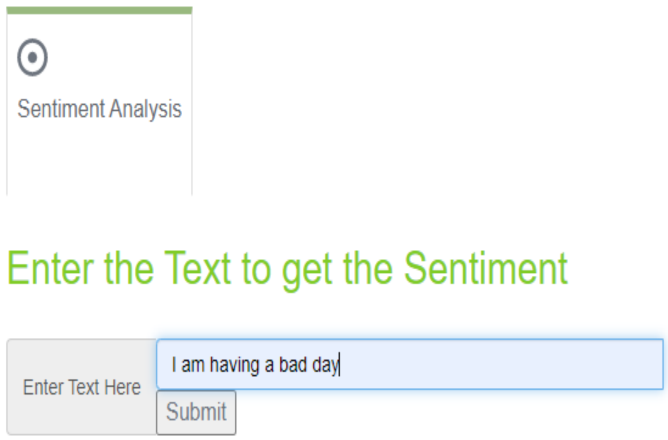

Fig 7Text is entered to check Sentiment

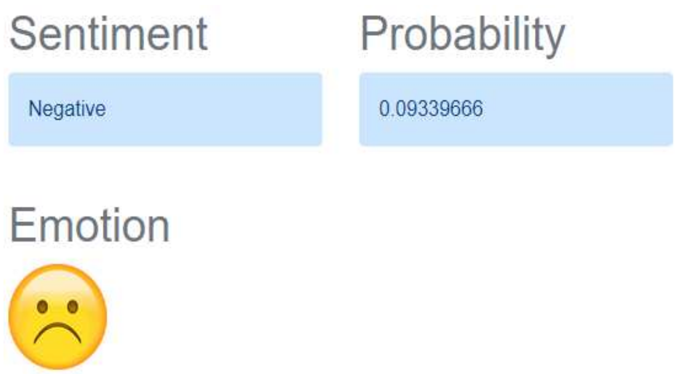

Fig 8 output is generated

This is the output of for the project. In fig 5 and fig 7 we have to add the statement for which we have to find the sentiment. The output for the above statement is displayed in fig 6 and fig 8, the above figure tells us whether the statement is negative or positive along with the sentiment associated with that, followed by an emoji. Probability depicts the strength of a negative or positive statement, if the probability is close to zero, it implies that the sentiment is strongly negative and if probability is close to one, it means that the statement is strongly positive.

VI. CONCLUSION

This model proposes sentiment classifiers that aids in the classification of emotion in text sequences. The model will predict whether the given text is positive or negative based on the user input. This model was created using LSTM, and the results showed that the Long Short Term Memory Networks algorithmic standard outperformed others in terms of precision.

Sentiment analysis is important since it allows companies to easily consider their consumers' overall views. By automatically classifying the sentiment behind reviews, social media conversations, and other data, you can make faster and more precise decisions. According to estimates, 90% of the world's data is unstructured or unorganised. Every day, massive amounts of unstructured business data are produced (emails, support tickets, talks, social media conversations, surveys, posts, documents, and so on). However, analysing opinion in a timely and productive manner is difficult.

Sentiment analysis can be applied to countless aspects of business, from brand monitoring and product analytics, to customer service and market research. By integrating it into their existing systems and analytics, leading brands are able to work faster, with more accuracy, toward more useful ends.

VII. FUTURE SCOPE

We can improve the accuracy using hyper parameter tuning on this particular neural network model. The above model predicts the sentiment of a single sentence; in the future, data in csv format will be given, and you can simply tweak it in the application. We plan to expand this research in the future so that various embedding models can be considered on a wider range of datasets.

ACKNOWLEDGEMENT

I should convey my obligation to Dr MN Nachappa and Prof: Dr SK Manju Bargavi and undertaking facilitators for their effective motivation and encouragement throughout my assessment work. Their ideal bearing, absolute co-action and second discernment have made my work gainful.

REFERENCES

[1] Zhang, Y. Wallace and Byron, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820,* 2015.

[2] T. a. C. K. a. C. G. a. D. J. Mikolov, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781,* 2013.

[3] J. L. Elman, "Finding structure in time," *Cognitive science,* 1990.

[4] S. a. S. J. Hochreiter, "Long short-term memory," *Neural computation,* 1997.

[5] Huang, G.-B. Zhu, Q.-Y. Siew and Chee-Kheong, "Extreme learning machine: theory and applications," *Neurocomputing,* 2016.

[6] Y. Wenpeng, K. Katharina and Y. M. S. Hinrich, "Comparative study of CNN and RNN for natural language processing," *arXiv preprint arXiv:1702.01923,* 2017.

[7] B. Yoshua, F. Paolo, u. Schmidhuber and E. Corso, "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term D ependencies* Sepp Hochreiter Fakult at f ur Informatik," *Citeseer,* 2001.

[8] Pennington, J. Socher, R. Manning and C. D, Glove: Global vectors for word representation, 2014.

[9] M. A. Daly, Raymond E Pham, P. T, H. Dan, N. A. Y and P. Christopher, Learning word vectors for sentiment analysis, 2011.

[10] Deriu, J. Milan and C. Mark, Sentiment analysis using convolutional neural networks with multi-task training and distant supervision on italian tweets, 2016.

[11] K. S, S. Kumar, M. Anand and S. KP, Sentiment analysis of tweets in malayalam using long short-term memory units and convolutional neural nets, 2017.